Normalization of Malay Noisy Text in Social Media using Levenshtein Distance and Rule-Based Techniques

***Azilawati Azizan[1], Nur Husna Anuar[2], Nurkhairizan Khairuddin[1], Rohana Ismail[3]**

[1]College of Computing, Informatics and Mathematics, Universiti Teknologi MARA (UiTM), Perak Branch, Tapah Campus, Malaysia.

[2]Yayasan Warisan Anak Selangor, Syarikat Pengurusan Projek TAWAS, Kompleks Belia & Kebudayaan Negeri Selangor, Shah Alam Selangor, Malaysia.

[3]Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Besut Campus, Terengganu, Malaysia.

DOI: https://dx.doi.org/10.47772/IJRISS.2024.8090125

Received: 01 September 2024; Accepted: 10 September 2024; Published: 08 October 2024

ABSTRACT

The rise of digital communication via hand phone and Internet has led to the widespread use of short-form words and abbreviations in text messaging. This trend poses challenges for data mining activities involving text processing and analysis, particularly in social media platforms where users employ a wide variety of abbreviations, slang, misspellings, and grammatical errors. To address this challenge, this study aimed to develop an algorithm for normalizing Malay noisy text using Levenshtein Distance (LD) and rule-based techniques. The LD is used to transform Malay spelling error words into their standard form, while rule-based techniques enhanced the conversion success rate for three categories of noisy term, namely slang, common Malay noisy text, and mixed language. The project was implemented using Python programming language, which demonstrated the effectiveness of the LD and rule-based techniques in normalizing noisy text in social media. The approach successfully normalized 80% of Malay noisy text into their standard text, which provides strong foundation for further study. Furthermore, this work open opportunities for introducing new approaches and rules to improve the normalization success rate, which can facilitate the analysis of text data in social media platforms. It is recommended that future studies focus on expanding the dataset and applying statistical validation methods to ensure the robustness and accuracy of the normalization model.

Keywords: Levenshtein Distance, Malay noisy text, Text normalization, Rule-based, Data mining

INTRODUCTION

The high prevalence of abbreviations, slang, and misspellings in Malay social media text impedes the accuracy of data mining activities such as sentiment analysis. The educational gap lies in the lack of comprehensive algorithms to handle such noisy text effectively. In data mining activities, especially those involving text processing such as Natural Language Processing (NLP), the accuracy and standardization of the words are crucial. Non-standard words, such as abbreviations, misspellings, missing punctuation, and slang, are categorized as noisy text and are considered meaningless for processing. While the traditional way has been on removing noise entirely, some researchers have argued that certain types of noise can actually carry valuable information (Sharou et al., 2021). For instance, in sentiment analysis, the presence of punctuation patterns or emojis may convey important emotional cues that should not be overlooked.

Current situation, social media is rife with noisy text (Hani, Nashaat, & Ahmed, 2019) due to user's tendency to use "text language" and abbreviate words, making it difficult to analyze the information. Different social media platforms have distinct features that have led to various writing styles among users (Hassan & Menezes, 2013). For example, SMS expands the nature of message shortening to avoid keystrokes, Facebook and instant messaging focus on emotional expression, and Twitter combines SMS and Facebook' features. These varying

genres of social media have resulted in diverse writing styles among social media users, such as repeating letters or punctuation for emphasis and emotional expression, exemplified by transforming of "good morning" to "goooood morniiing."

Moreover, the widespread use of phonetic spelling reflects the local accents, such as "wuz up bro" instead of "what is up brother." In addition, users often eliminate vowels, as in "cm to c my luv" instead of "come to see my love." Users may also substitute numbers for letters, such as "4get" for "forget," "2morrow" for "tomorrow," or "b4" for "before," and even replace certain letters with phonetically similar ones, such as "fon" for "phone." Slang abbreviations that shorten multi-word expressions are also common, such as "LMS" for "like my status," "idk" for "I do not know," or "ROFL" for "rolling on floor laughing."

Figure 1 shows examples of text communication on Twitter and Facebook. Misunderstandings are more likely to occur when users opt for abbreviated or otherwise non-standard forms of communication, sacrificing clarity for ease of use. Non-standard text often appears unfamiliar to machines and can lead to misinterpretation of the intended message, in fact it can also cause misunderstanding among humans. However, users believe that using non-standard text can speed up their typing and provide a convenient way to express their feelings through a keyboard.

Figure 1: Example of communication in Twitter and Facebook

As a result, a significant amount of social media text cannot be translated or may be misinterpreted leading to loss of information. Noisy words in text creates challenges for software or applications (Saloot, Idris, & Mahmud, 2014) attempting to extract the true meaning of the content, particularly in sentiment analysis of User Generated Content (UGC) (Han, Wang, Zhang, & Wang, 2020). Pre-processing noisy and standard text is fundamentally different since noisy text lacks of standard rules or patterns, whereas standard text follows the conventional language format.

Noisy text can be classified into several different types, namely incorrect abbreviations, incorrect spellings, nonstandard terminology, missing punctuation and use of slang's (Samsudin, Puteh, Razak, & Zakree, 2012). Additional types include false starts, repetitions, missing letter case information, pause-filling words, phonetic substitutions and unstructured grammar (Desai & Narvekar, 2015). Table 1 shows some types of noisy terms, and examples of commonly used noisy text on social media.

Table 1: List of Common Malay Short-form and its Full Word

No	Types of noisy text	Texting language	Standard language
1.	Spelling Error	mekdi	McDonald
2.	Abbreviation/ Short Form	mlm	Malam
3.	Slang	hang	Kamu
4.	Phonetic Text	se7	Setuju
5.	Mix-Code Language	bestnya	Seronoknya
6.	Symbol	&	And
7.	Emoticon	:)	Happy
8.	Repeating Word	Sama2	Sama-sama

Social media platforms are rife with these types noisy texts (as in Table 1), which poses significant challenges for text mining and natural language processing, as traditional algorithms struggle with the lack of standardization in these texts. While there has been research on text normalization, much of it has focused on English or other widely spoken languages. Therefore, this study addresses the challenge of noisy Malay text normalization in user-generated content on social media. The high prevalence of abbreviations, slang, and misspellings in Malay social media text impedes the accuracy of data mining activities such as sentiment analysis. The educational gap lies in the lack of comprehensive algorithms to handle such noisy text effectively. This study aims to (i) develop a hybrid algorithm using Levenshtein Distance and rule-based techniques for normalizing noisy Malay text and (ii) evaluate the algorithm's effectiveness in comparison to traditional normalization methods. The research questions are; how effective is the proposed algorithm in normalizing different types of Malay noisy text? Can combining rule-based techniques with LD improve the overall success rate?

RELATED WORK

Recent research has concentrated on improving text analysis by developing effective normalization techniques that maintain valuable information in noisy text. For example, Khan & Lee (2021) proposed a hybrid normalization technique to retain essential information in social media text rather than filtering it out. Lourentzou et al. (2019) stressed the significance of contextual information in text normalization and introduced a hybrid word-character attention-based encoder-decoder model for social media text normalization.

Levenshtein Distance (LD)

Levenshtein distance (or edit distance) is a metric that measures the similarity between two strings. It is named after the Soviet mathematician Vladimir Levenshtein (Yang, Zeng, Fu, & Luo, 2020), who introduced this concept in 1965. LD is commonly used in various applications such as spell checking (Santoso, Yuliawati, Shalahuddin, & Wibawa, 2019), search engine algorithms, machine translation (Po, 2020), and speech recognition (Contreras, Ayala, & Cruz, 2020). It is also employed in DNA sequencing and analysis, as well as in data mining and text data pre-processing tasks such as text normalization and text cleaning (Riza, Syaiful Anwar, Rahman, Abdullah, & Nazir, 2020). Additionally, LD can be used in similarity analysis applications, such as plagiarism detection and image analysis (Ounachad, 2020).

The LD method is a popular technique used in text normalization (Bollmann, 2019). It measures the similarity between two strings by calculating the minimum number of operations required to transform one string into the other (Mehta, Salgond, Satra, & Sharma, 2021). These operations include insertion, deletion, and substitution of characters. For example, if the strings x='past' and y='past' are compared, the LD value will be 0 because no transformation is needed as the strings are already identical (LD (x,y) = 0). On the other hand, if x='past' is compared to y='last', the LD value will be 1 as only one substitution (replacing 'p' with 'l') is required to transform 'past' into 'last' (LD (x,y) = 1). The LD value increases with the degree of difference between the two strings. This implies that when strings are more different, the LD value will be greater, whereas when they are more similar, the LD value will be smaller. By using the LD method, text normalization can be achieved efficiently, ensuring that noisy and non-standard texts are converted into their standard forms for accurate processing and analysis.

Rule-Based Technique

Rule-based techniques are another way to normalize noisy text in social media (Chakraborty et al., 2020). These techniques rely on a set of pre-defined rules to transform noisy text into a standardized form. Rule-based techniques offer a structured and systematic way to identify and address common types of noise, such as spelling errors, grammatical mistakes, and formatting inconsistencies. Kaur & Singh (2020) developed a rule-based Roman to Gurmukhi text normalization system, highlighting the challenge of capturing the noisy nature of data. Additionally, Tursun & Cakici (2017) utilized the noisy channel model method for normalizing non-standard words in Uyghur text, showcasing the importance of analyzing large corpora of noisy and formal texts. By leveraging a set of predefined rules, these techniques can effectively clean and normalize text data, improving the overall quality and reliability of the input for downstream tasks. Some common rule-based techniques used for text normalization are:

Look-Up Tables

Look-up tables are pre-defined tables containing mappings between noisy text and its normalized form. These tables may include typical acronyms, abbreviation, slang terms, or typos that are frequently found on social media platforms. When the system encounters a noisy term, it looks up the term in the table and replaces it with the corresponding normalized form (Barman, Sarmah, & Sarma, 2020).

Stemming

It is a technique for reducing a word to its root form. For instance, "running" and "runner" can be stemmed to "run". This technique can help to increase the accuracy of text analysis and decrease the number of unique terms in a corpus of texts (Barman et al., 2020).

Part-of-Speech Tagging

It is a process of assigning a part of speech (noun, verb, adjective, etc.) to each word in a sentence (Li, Mao, & Wang, 2022). This approach can identify and correct grammatical mistakes in writing (Pham, 2020).

Regular Expressions

Regular expressions are patterns used to match character combinations in a string and manipulate text (Borsotti, Breveglieri, Crespi, & Morzenti, 2023). They are frequently used to remove unwanted characters, punctuation, or symbols from text (Topaz et al., 2019). For instance, a regular expression can replace all instances of an abbreviation and acronym with its full form. By specifying the pattern of the noisy text, it is possible to replace or remove the pattern using regex. Regular expressions can be very powerful and flexible, but they require knowledge of the specific patterns in the noisy text that need to be normalized. They can also be computationally expensive when dealing with large amounts of text.

In conclusion, recent research in the field of noisy text normalization has focused on developing hybrid techniques, leveraging contextual information, and utilizing rule-based approaches to address the challenges posed by noisy text, particularly from sources like social media.

IMPLEMENTATION IN THIS STUDY

This study employs a quantitative approach. It involves the collection and analysis of numerical data (the success rate of text normalization). The Levenshtein Distance and rule-based techniques were applied to a dataset of 1,000 Malay-language comments, and the success rate of normalization was measured quantitatively. A rule-based algorithm was employed to normalize noisy text that includes common Malay short-form words, mixed language, and slang. The primary rule served as a reference for common short-form words in Malay. If a noisy text was found in the list of common short-form words, it was replaced with its standard word. For instance, the short-form word 'tak' was replaced with its standard word 'tidak'. A total of 130 common Malay short-form terms and their corresponding standard words from research articles and personal observations were compiled. Table 2 depicts the list of common Malay short-form words and their standard words.

In addition, new rule-based algorithms were created to normalize northern (Kedah) slang and regional (Kuala Lumpur) slang. Typically, regional slang can be identified when a noisy text ends with the character ‘e’, such as ‘bile’, ‘mane’, ‘siape’, and ‘kite’. On the other hand, northern slang is usually identified when the noisy text ends with the character ‘q’, such as ‘lapaq’, ‘pasaq’, ‘bayaq’, and ’tawaq’. Noisy text that end with the character ‘e’ are replaced with the character ‘a’, while those that end with the character ‘q’ are replaced with the character ‘r’.

Table 2: List of Common Malay Short-form and its Full Word

	Short Form	Full Word	Short Form	Full Word
2	x	tidak	die	dia
3	g	pergi	leh	boleh
4	p	pergi	lak	pulak
5	k	okay	pe	apa
6	d	di	pew	apa
7	aq	aku	ig	instagram
8	ko	engkau	yt	youtube
9	kt	dekat	fb	facebook
10	nie	ini	gak	juga
11	ne	mana	suma	semua
12	dk	duduk	aritu	hari itu
13	kat	dekat	arini	hari ini
14	de	ada	mekdi	mcd
15	gi	pergi	kepci	kfc
16	pegi	pergi	len	lain
17	sy	saya	kl	kuala lupur
18	bg	bagi	dgn	dengan
19	tp	tapi	abg	abang
20	ja	saja	korang	kau orang
21	je	saja	jgk	juga

Furthermore, two methods were proposed to normalize the text that mix English and Malay words. Based on our observations, such words can typically be identified when the English word ends with the character ‘la’ or ‘nya’, for instance, ‘goodnya’, ‘seriusla’, ‘confidentnya’, and ‘todayla’. For the first method, the characters ‘la’ or ‘nya’ were removed to normalize the noisy text. The second method involved checking the noisy text against a list of mixed language words, and replacing it with its standard word. For example, the mixed language word ‘kompom’ is replaced with its standard word ‘confirm’.

METHODOLOGY

For the development, the Rapid Application Development (RAD) methodology was utilized. RAD emphasizes the rapid development of applications through frequent iterations and continuous feedback. Figure 2 illustrates the main phases involved in the RAD methodology.

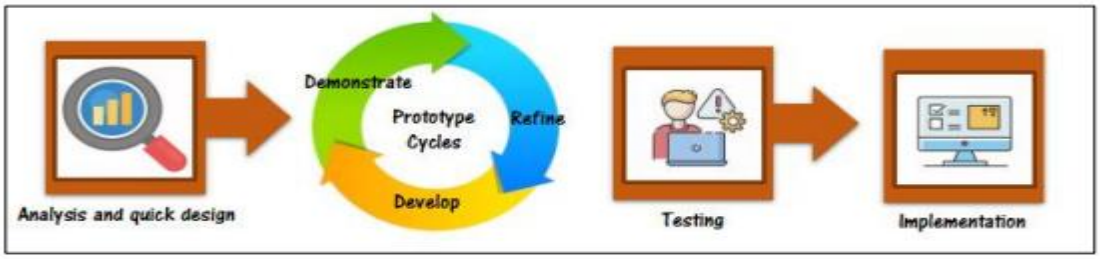

Figure 2: Rapid Application Development (RAD) Phases

During the initial phase (analysis and quick design), information pertaining to Malay noisy text was gathered through social media observation and reading of articles and journals. This information was used to compile a list of noisy text types and examples, which were stored in a database for comparison between standard and non-standard text. The identification and normalization process flow, as well as the main user interface, were also designed during this phase. In the subsequent phase (prototype cycle), a normalization application prototype was developed using the Python programming language. The prototype was then tested using selected test data to evaluate its initial performance. After several refinement processes, the prototype was deemed ready for full implementation. To abstract overall flow of the application, the system architecture is illustrated in Figure 3.

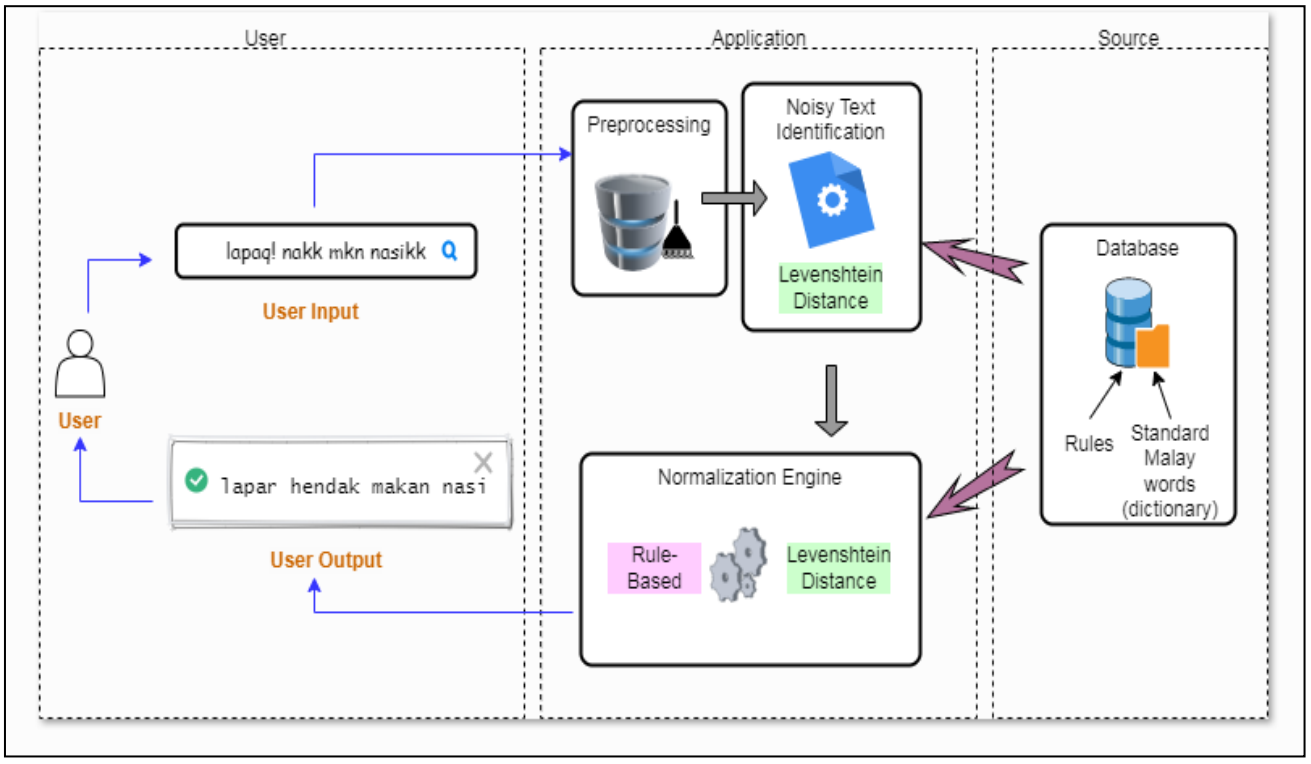

Figure 3: System Architecture

The architecture comprises three primary blocks.

i. **User Block**: Includes the user, user input, and user output. In this block, the user inputs text or sentences and then clicks the normalize button.

ii. **Application Block**: In this block, the input text or sentence is pre-processed and tokenized, breaking it down into individual tokens. Each token is identified as either standard text or noisy text using Levenshtein Distance (LD) and a list of standard Malay words from the database. Table 3 shows a sample of noisy words compared to their standard words and the LD values.

iii. **Source Block**: Contains the rules and a list of standard texts for comparison to determine whether the text is noisy or not.

Table 3: Standard Malay Dictionary Words, Noisy Words and LD Values

	Dictionary	Noisy Words	Levenshtein
1023	agar	dgar	1
3250	ambil	ambik	1
9171	bateri	batteri	1
10922	dan	dah	1
10924	dan	dgn	1
12416	expensive	expemsive	1
12505	cakap	ckp	2
13358	harga	hrga	1
15099	rama	rama2	1
15467	pagi	pegi	1

Once all tokens are identified, the normalization process is executed, and a list of suggested standard texts is displayed on the user output screen as in Figure 4.

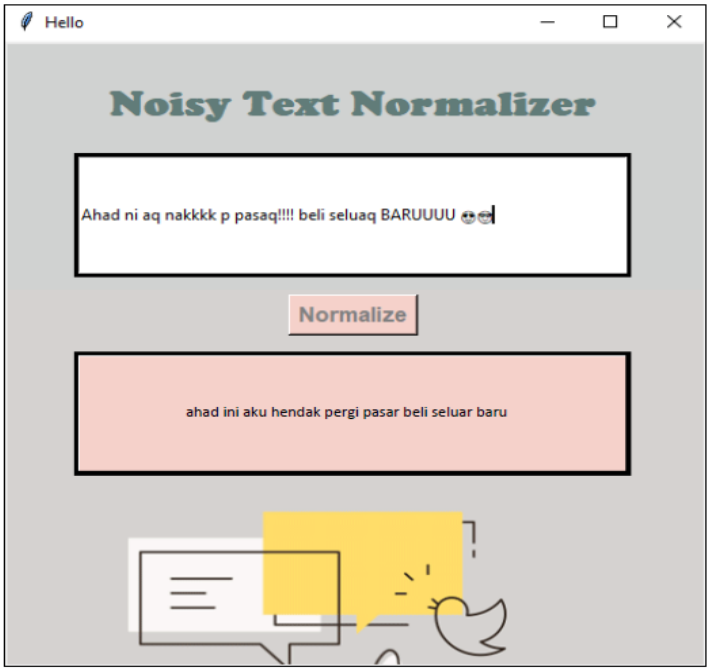

Figure 4: The User Interface of the Application

The user interface is designed to be user-friendly and intuitive, enabling users to easily input text, view normalization suggestions, and select the appropriate standardized text. The interface includes the following features:

Text Input Field: Allows users to enter the text or sentences they want to normalize.

Normalize Button: Users click this button to initiate the normalization process.

Output Display: Shows the suggested normalized text.

RESULT AND DISCUSSION

Initially, 1000 comments from YouTube containing a mix of noisy and standard language were gathered. After

the data cleaning process, only 100 comments were chosen to test the application comprising 1216 words, including 701 (58%) standard texts and 515 (42%) noisy texts.

The noisy texts were then normalized into standard text using the LD method. Of these, 228 were successfully normalized and 287 were not. The failures were due to the wide distance between the standard words. For example, the word ‘cane’ is a short form of ‘macam mana’ but, LD normalized it to ‘aneh’ because the distance from "cane" to "macam mana" is 7, while from "cane" to "aneh" it is only 2. However, when using LD combined with the rule-based normalization, 410 out of 515 noisy texts were correctly normalized, achieving 80% of success rate.

The combination of several new rule-based algorithms with LD provides a significant improvement in normalization result. While LD can directly normalize certain types of noisy texts, such as spelling errors, many noisy texts require additional rule-based techniques to be accurately normalized into standard text. Figure 5 shows the comparison between the approach of using only LD and the method that combines LD with rule-based techniques. The results indicates that when the new rule-based are combined with the LD method, the performance improves significantly, achieving up to 80% accuracy compared to 44% when LD is used to normalize the noisy text.

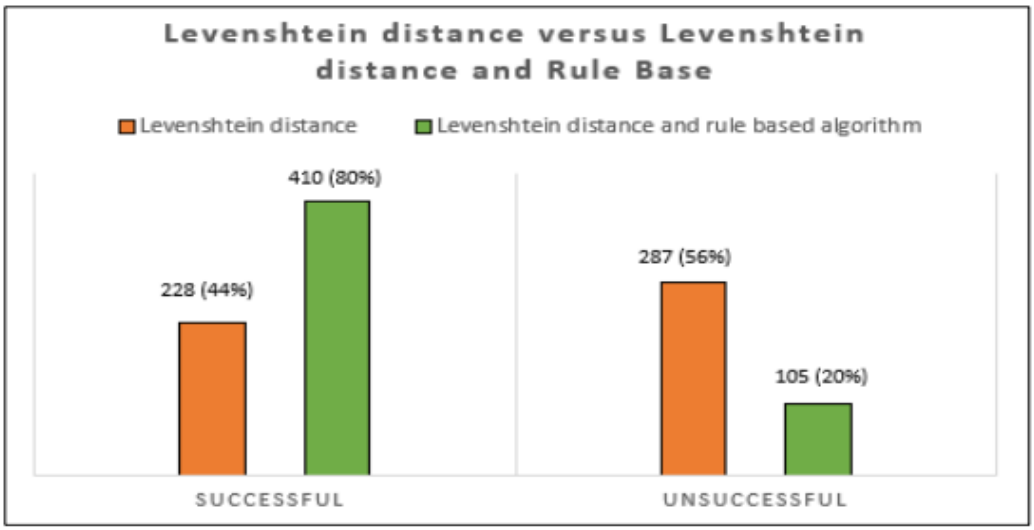

Figure 5: Comparison between Two Method

In the analysis of our text normalization techniques, we observed that using LD alone resulted in an unsuccessful normalization rate of 56%. This suggests that while LD can be effective, it struggles with a significant proportion of noisy text cases. However, when we combined LD with a rule-based approach, the unsuccessful rate dropped significantly to 20%. This improvement indicates that integrating rule-based methods with LD can enhance the overall accuracy of text normalization, reducing the number of cases where the technique fails to normalize effectively.

LIMITATION AND RECOMMENDATION

The study addresses the challenges posed by noisy Malay text sourced from social media, which has been found to impede text processing and leads to erroneous results in text mining activities. However, the implementation of some rules is limited by the reference to a list of the 130 most common Malay short form words. More rules could be produced if more short form words are identified. The proposed slang rule is limited to the northern and regional slang in Malaysia. Given that this study covers only certain categories of Malay noisy texts. Further rules (regular expressions) could be incorporated to enhance the application in the future. Notably, the inclusion of rules for other regions like the east coast, south, or East Malaysia (Sabah and Sarawak), holds promise for future development of the proposed application. In addition, incorporating machine learning techniques, such as deep learning, to enhance the rule-based system's adaptability to new noisy text patterns and applying statistical validation techniques to future studies to further authenticate the

findings are also recommended.

CONCLUSION

Noisy text is a significant challenge that can lead to misunderstanding, information loss, and other issues, particularly in text mining activities. This problem is present in Malay text from social media, which is notoriously difficult to process due to the high prevalence of noisy text. This study focuses on identifying Malay noisy text using LD and developing an algorithm that utilizes multiple rules-based approaches to normalize common Malay noisy text into standard word.

The proposed application was tested using 100 Malay-language comments on YouTube and achieved 80% of success rate. However, this project represents only a starting point, as numerous rules related to Malay noisy text that remain unexplored. Processing Malay text from social media is an arduous, given the abundance of noisy text, and further research in this area has significant potential to benefit the field of text processing.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the Universiti Teknologi MARA (UiTM), Perak branch for providing full support in the completion of this project.

REFERENCES

1. Barman, A. K., Sarmah, J., & Sarma, S. K. (2020). Development of assamese rule based stemmer using WordNet. Proceedings of the 10th Global WordNet Conference, 135–139.
2. Bollmann, M. (2019). A large-scale comparison of historical text normalization systems. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1, 3885–3898. https://doi.org/10.18653/v1/n19-1389
3. Borsotti, A., Breveglieri, L., Crespi, S., & Morzenti, A. (2023). General parsing with regular expression matching. Journal of Computer Languages, 74(March 2022), 101176. https://doi.org/10.1016/j.cola.2022.101176
4. Chakraborty, K., Bhatia, S., Bhattacharyya, S., Platos, J., Bag, R., & Hassanien, A. E. (2020). Sentiment Analysis of COVID-19 tweets by Deep Learning Classifiers—A study to show how popularity is affecting accuracy in social media. Applied Soft Computing Journal, 97, 106754. https://doi.org/10.1016/j.asoc.2020.106754
5. Contreras, R., Ayala, A., & Cruz, F. (2020). Unmanned aerial vehicle control through domain-based automatic speech recognition. Computers, 9(3), 1–15. https://doi.org/10.3390/computers9030075
6. Desai, N., & Narvekar, M. (2015). Normalization of noisy text data. Procedia Computer Science, 45(C), 127–132. https://doi.org/10.1016/j.procs.2015.03.104
7. Han, X., Wang, J., Zhang, M., & Wang, X. (2020). Using Social Media to Mine and Analyze Public Opinion Related to COVID-19 in China.
8. Hani, J., Nashaat, M., & Ahmed, M. (2019). Social Media Cyberbullying Detection using Machine Learning, (May). https://doi.org/10.14569/IJACSA.2019.0100587
9. Hassan, H., & Menezes, A. (2013). Social text normalization using contextual graph Random Walks. ACL 2013 - 51st Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, 1, 1577–1586.
10. Li, H., Mao, H., & Wang, J. (2022). Part-of-speech tagging with rule-based data preprocessing and transformer. Electronics (Switzerland), 11(1). https://doi.org/10.3390/electronics11010056
11. Mehta, A., Salgond, V., Satra, D., & Sharma, N. (2021). Spell Correction and Suggestion Using Levenshtein Distance. International Research Journal of Engineering and Technology, 1977–1981. Retrieved from www.irjet.net
12. Ounachad, K. (2020). Human Face (Sketch/Photo) Age Group Estimation and Classification Using Perfect Face Ratios and Levenshtein Distance. International Journal of Emerging Trends in Engineering Research, 8(7), 3191–3201. https://doi.org/10.30534/ijeter/2020/52872020

13. Pham, B. (2020). Parts of Speech Tagging: Rule-Based Parts of Speech Tagging: Rule-Based Parts of Speech Tagging: Rule-Based. Retrieved from https://digitalcommons.harrisburgu.edu/cisc_student-coursework
14. Po, D. K. (2020). Similarity Based Information Retrieval Using Levenshtein Distance Algorithm. International Journal of Advances in Scientific Research and Engineering, 06(04), 06–10. https://doi.org/10.31695/ijasre.2020.33780
15. Riza, L. S., Syaiful Anwar, F., Rahman, E. F., Abdullah, C. U., & Nazir, S. (2020). Natural Language Processing and Levenshtein Distance for Generating Error Identification Typed Questions on TOEFL Journal of Computers for Society. Jcs, 1(1), 1–23.
16. Saloot, M. A., Idris, N., & Mahmud, R. (2014). An architecture for Malay Tweet normalization. Information Processing and Management, 50(5), 621–633. https://doi.org/10.1016/j.ipm.2014.04.009
17. Samsudin, N., Puteh, M., Razak, A., & Zakree, M. (2012). N_ormalization of Common N_oisyTerms in Malaysian Online Media. In Proceedings of the Knowledge Management International Conference, (July), 515–520. Retrieved from http://www.kmice.cms.net.my/ProcKMICe/KMICe2012/PDF/CR204.pdf
18. Santoso, P., Yuliawati, P., Shalahuddin, R., & Wibawa, A. P. (2019). Damerau Levenshtein Distance for Indonesian Spelling Correction. Jurnal Informatika, 13(2), 11. https://doi.org/10.26555/jifo.v13i2.a15698
19. Topaz, M., Murga, L., Gaddis, K. M., McDonald, M. V., Bar-Bachar, O., Goldberg, Y., & Bowles, K. H. (2019). Mining fall-related information in clinical notes: Comparison of rule-based and novel word embedding-based machine learning approaches. Journal of Biomedical Informatics, 90(June 2018), 103103. https://doi.org/10.1016/j.jbi.2019.103103
20. Yang, L., Zeng, Y., Fu, S., & Luo, Y. (2020). Unsupervised analysis of encrypted video traffic based on levenshtein distance. Communications in Computer and Information Science (Vol. 1298 CCIS). Springer Singapore. https://doi.org/10.1007/978-981-15-9031-39